Assignment #7

Jingye Wang

October 17, 2016

```
setwd("~/Dropbox/WUSTL third/Multilevel Modeling for Quantitative
Research/assignment/7")
.libPaths("/Library/Frameworks/R.framework/Versions/3.3/Resources/library")
```

Ch16.3

Using the data in folder cd4 regarding CD4 percentages for young children with HIV, we shall revisit Exercise 12.2.

```
library(rjags)
library(R2jags)
library(R2WinBUGS)
library(dplyr)
```

Part A

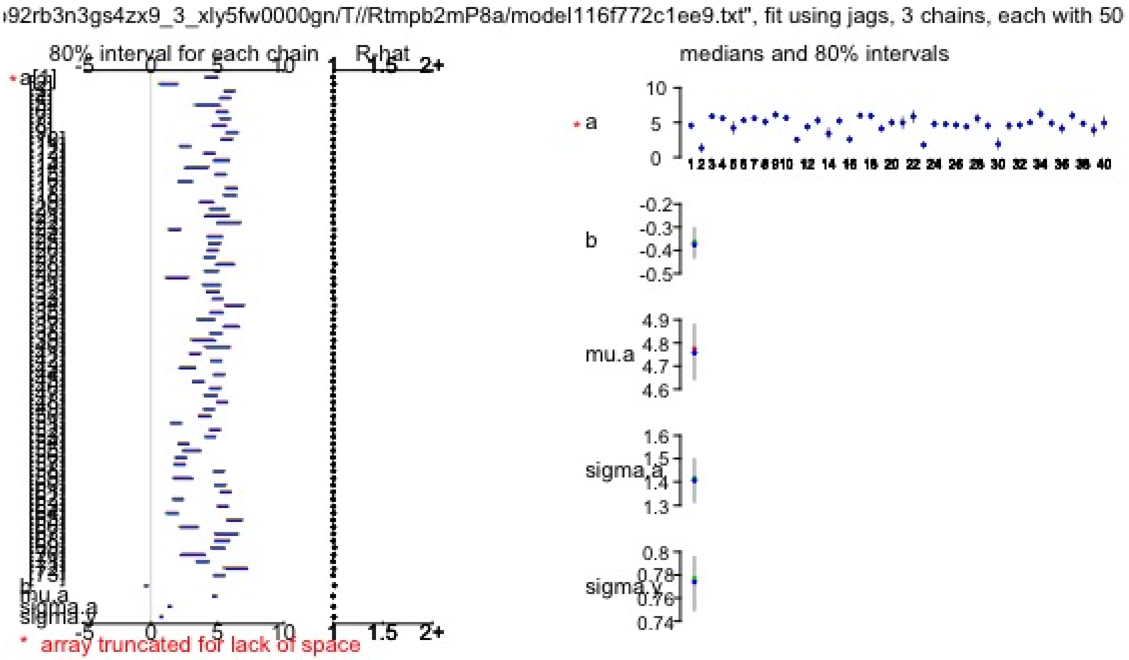
Use Bugs to fit the model in Exercise 12.2(a). Interpret the results.

```
# import data
hiv_data <- read.csv ("allvar.csv", header= T)
hiv_data$time <- hiv_data$visage - hiv_data$baseage
hiv_data_1 <- hiv_data[,c(2,4,10)]
hiv_data_1 <- na.omit(hiv_data_1)

# using JAGS
CD4PCT <- hiv_data_1$CD4PCT
n <- length(CD4PCT)
y <- sqrt(CD4PCT)
x <- hiv_data_1$time
id.name <- as.vector(hiv_data_1$newpid)
uniq <- unique(id.name)
J <- length(uniq)
id <- rep (NA, J)
for (i in 1:J){
  id[id.name==uniq[i]] <- i
}

# model
model_1 <- function() {
  for (i in 1:n){
    y[i] ~ dnorm (y.hat[i], tau.y)
    y.hat[i] <- a[id[i]] + b*x[i]
  }
  b ~ dnorm (0, .0001)
  tau.y <- pow(sigma.y, -2)
  sigma.y ~ dunif (0, 100)

  for (j in 1:J){
    a[j] ~ dnorm (mu.a, tau.a)
  }
  mu.a ~ dnorm (0, .0001)
  tau.a <- pow(sigma.a, -2)
  sigma.a ~ dunif (0, 100)
}

# hiv.inits and hiv.parameters
hiv_data_1_ls <- list ("n", "J", "y", "id", "x")
hiv.inits <- function (){
list (a=rnorm(J),
      b=rnorm(1),
      mu.a=rnorm(1),
      sigma.y=runif(1),
      sigma.a=runif(1))
  }
hiv.parameters <- c ("a", "b", "mu.a", "sigma.y", "sigma.a")

# run the model
hiv_bugs_1 <- jags(hiv_data_1_ls, hiv.inits, hiv.parameters, model_1, n.chains=3, n.iter
=500, DIC=F)
hiv_bugs_1 %>% plot
hiv_bugs_1$BUGSoutput
```

Output

```
Inference for Bugs model at "/var/folders/ld/tp92rb3n3gs4zx9_3_xly5fw0000gn/T//Rtmpb2mP8
a/model116f772c1ee9.txt", fit using jags,
 3 chains, each with 500 iterations (first 250 discarded)
 n.sims = 750 iterations saved
        mean  sd 2.5%  25%  50%  75% 97.5% Rhat n.eff
a[1]     4.5 0.3  3.8  4.3  4.6  4.8   5.2    1   750
a[2]     1.3 0.5  0.3  1.0  1.3  1.6   2.3    1   400
a[3]     5.9 0.3  5.3  5.7  5.9  6.1   6.4    1   580
a[4]     5.6 0.3  5.0  5.4  5.6  5.8   6.1    1   750
a[5]     4.2 0.7  2.7  3.7  4.2  4.6   5.6    1   550
a[246]   5.7 0.5  4.7  5.3  5.7  6.0   6.6    1   410
a[247]   4.0 0.7  2.6  3.6  4.0  4.4   5.3    1   280
a[248]   4.6 0.5  3.5  4.2  4.6  4.9   5.6    1   370
a[249]   3.9 0.7  2.6  3.4  3.9  4.3   5.1    1   750
a[250]   3.9 0.5  2.9  3.5  3.9  4.2   4.8    1   750
b       -0.4 0.1 -0.5 -0.4 -0.4 -0.3  -0.3    1   750
mu.a     4.8 0.1  4.6  4.7  4.8  4.8   4.9    1   750
sigma.a  1.4 0.1  1.3  1.4  1.4  1.5   1.6    1   420
sigma.y  0.8 0.0  0.7  0.8  0.8  0.8   0.8    1   300

For each parameter, n.eff is a crude measure of effective sample size,
and Rhat is the potential scale reduction factor (at convergence, Rhat=1).
```

Graph



Based on this model, b=-0.4. In this model, each increase of 1 in time corresponds to a 0.4 predicted decrease in the square root scale of CD4 percentage. In Ch12.2a, the fixed effect of time is -0.36609. Each increase of 1 in time corresponds to a 0.37 predicted decrease in the square root scale of CD4 percentage.

Part B

Use Bugs to fit the model in Exercise 12.2(b). Interpret the results.

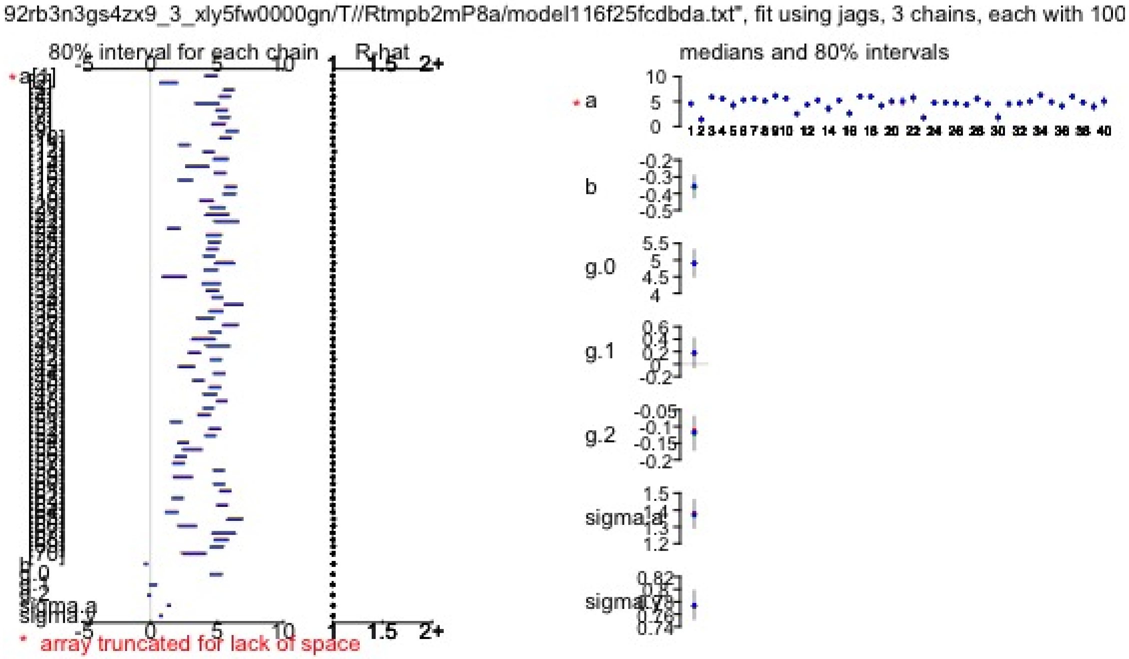
```
hiv_data_2 <- hiv_data[,c('newpid', 'CD4PCT', 'treatmnt', 'baseage', 'time')]
hiv_data_2 <- na.omit(hiv_data_2)

# using JAGS
CD4PCT <- hiv_data_2$CD4PCT
n <- length(CD4PCT)
y <- sqrt(CD4PCT)
x <- hiv_data_2$time
P <- hiv_data_2[, c('newpid', 'treatmnt', 'baseage')] %>% unique %>% select(., -(newpi
d))
id.name <- as.vector(hiv_data_2$newpid)
uniq <- unique(id.name)
J <- length(uniq)
id <- rep (NA, J)
for (i in 1:J){
  id[id.name==uniq[i]] <- i
}

# hiv.inits and hiv.parameters
hiv_data_list_2 <- list ("n", "J", "y", "id", "x", 'P')
hiv.inits_2 <- function () {
  list (a=rnorm(J),
        b=rnorm(1),
        sigma.y=runif(1),
        sigma.a=runif(1),
        g.0=rnorm(1),
        g.1=rnorm(1),
        g.2=rnorm(1))
}
hiv.parameters_2 <- c ("a", "b", "sigma.y", 'g.0', 'g.1', 'g.2', "sigma.a")

# model
model_2 <- function() {
  for (i in 1:n){
    y[i] ~ dnorm (y.hat[i], tau.y)
    y.hat[i] <- a[id[i]] + b*x[i]
  }
  b ~ dnorm (0, .0001)
  tau.y <- pow(sigma.y, -2)
  sigma.y ~ dunif (0, 100)

  for (j in 1:J){
    a[j] ~ dnorm (a.hat[j], tau.a)
    a.hat[j] <- g.0 + g.1*P[j, 1] + g.2*P[j, 2]
  }
  g.0 ~ dnorm (0, .0001)
  g.1 ~ dnorm (0, .0001)
  g.2 ~ dnorm (0, .0001)
  tau.a <- pow(sigma.a, -2)
  sigma.a ~ dunif (0, 100)
}

# run the model
```

```
hiv_bugs_2 <- jags(hiv_data_list_2, hiv.inits_2, hiv.parameters_2, model_2, n.chains=3,
 n.iter=1000, DIC=F)
hiv_bugs_2 %>% plot
hiv_bugs_2$BUGSoutput
```

Output

```
Inference for Bugs model at "/var/folders/ld/tp92rb3n3gs4zx9_3_xly5fw0000gn/T//Rtmpb2mP8
a/model116f25fcdbda.txt", fit using jags,
 3 chains, each with 1000 iterations (first 500 discarded)
 n.sims = 1500 iterations saved
        mean  sd 2.5%  25%  50%  75% 97.5% Rhat n.eff
a[1]     4.5 0.3  3.9  4.3  4.5  4.8   5.2    1  1500
a[2]     1.4 0.5  0.4  1.0  1.4  1.7   2.4    1  1500
a[3]     5.9 0.3  5.3  5.7  5.9  6.1   6.4    1  1500
a[4]     5.6 0.3  5.0  5.4  5.6  5.8   6.2    1   520
a[5]     4.2 0.7  2.9  3.8  4.3  4.7   5.6    1  1500
a[245]   5.4 0.7  4.1  5.0  5.4  5.9   6.7    1  1500
a[246]   5.7 0.5  4.7  5.4  5.7  6.0   6.6    1  1500
a[247]   4.0 0.7  2.6  3.5  4.0  4.5   5.4    1  1500
a[248]   4.6 0.5  3.6  4.2  4.6  4.9   5.6    1   690
a[249]   3.9 0.7  2.6  3.5  3.9  4.4   5.3    1  1500
a[250]   3.9 0.5  3.0  3.6  3.9  4.2   4.9    1  1100
b       -0.4 0.1 -0.5 -0.4 -0.4 -0.3  -0.3    1   380
g.0      4.9 0.3  4.3  4.7  4.9  5.1   5.5    1  1500
g.1      0.2 0.2 -0.2  0.1  0.2  0.3   0.6    1  1500
g.2     -0.1 0.0 -0.2 -0.1 -0.1 -0.1   0.0    1  1300
sigma.a  1.4 0.1  1.2  1.3  1.4  1.4   1.5    1   540
sigma.y  0.8 0.0  0.7  0.8  0.8  0.8   0.8    1  1000

For each parameter, n.eff is a crude measure of effective sample size,
and Rhat is the potential scale reduction factor (at convergence, Rhat=1).
```

Graph



92rb3n3gs4zx9_3_xly5fw0000gn/T//Rtmpb2mP8a/model116f25fcdbda.txt", fit using jags, 3 chains, each with 100

Based on this model, b=-0.4, g.1=-0.2, g.2=-0.1. In this model, each increase of 1 in time corresponds to a 0.4 predicted decrease in the square root scale of CD4 percentage. Compared to treatment 1, treatment 2 corresponds to a 0.2 predicted increase in the square root scale of CD4 percentage. Each increase of 1 in baseline age corresponds to a 0.1 predicted decrease in the square root scale of CD4 percentage. In Ch12.2b, the fixed effect of time is -0.36216, treatment is 0.18008, and basement age is -0.11945. Each increase of 1 in time corresponds to a 0.36 predicted decrease in the square root scale of CD4 percentage. Compared to treatment 1, treatment 2 corresponds to a 0.18 predicted increase in the square root scale of CD4 percentage. Each increase of 1 in baseline age corresponds to a 0.12 predicted decrease in the square root scale of CD4 percentage.

Ch16.8

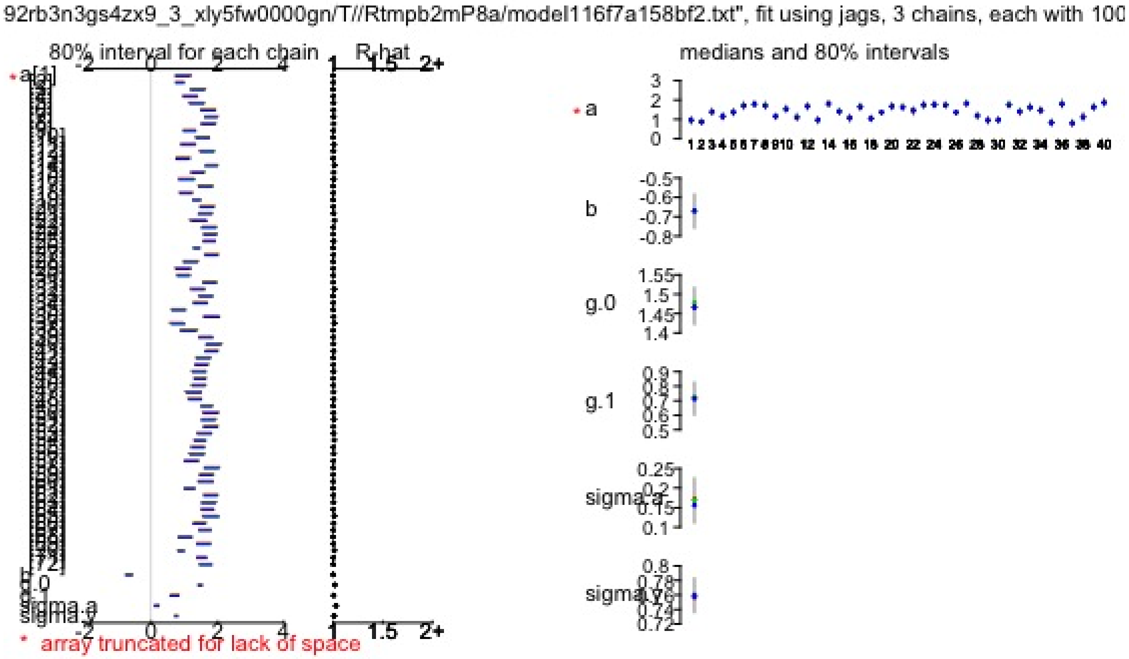
Impact of the prior distribution: you will use Bugs to fit several versions of the varying-intercept model to the radon data using floor as a house-level predictor and uranium as a county-level predictor.

```
# import data
srrs2 <- read.table ("srrs2.dat", header=T, sep=",")
mn <- srrs2$state=="MN"
radon <- srrs2$activity[mn]
log.radon <- log (ifelse (radon==0, .1, radon))
floor <- srrs2$floor[mn]       # 0 for basement, 1 for first floor
n <- length(radon)
y <- log.radon
x <- floor

# get county index variable
county.name <- as.vector(srrs2$county[mn])
uniq <- unique(county.name)
J <- length(uniq)
county <- rep (NA, J)
for (i in 1:J){
  county[county.name==uniq[i]] <- i
}

srrs2.fips <- srrs2$stfips*1000 + srrs2$cntyfips
cty <- read.table ("cty.dat", header=T, sep=",")
usa.fips <- 1000*cty[,"stfips"] + cty[,"ctfips"]
usa.rows <- match (unique(srrs2.fips[mn]), usa.fips)
uranium <- cty[usa.rows,"Uppm"]
u <- log (uranium)

# radon.data and radon.data
radon.data <- list ("n", "J", "x", "y", "county", "u")
radon.parameters <- c ("a", "b", "g.0", "g.1", "sigma.y", "sigma.a")
radon.inits <- function (){
  list (a=rnorm(J),
        b=rnorm(1),
        g.0=rnorm(1),
        g.1=rnorm(1),
        sigma.y=runif(1),
        sigma.a=runif(1))
}

# model
model_0 <- function () {
  for (i in 1:n){
    y[i] ~ dnorm (y.hat[i], tau.y)
    y.hat[i] <- a[county[i]] + b*x[i]
  }
  b ~ dnorm (0, .0001)
  tau.y <- pow(sigma.y, -2)
  sigma.y ~ dunif (0, 100)

  for (j in 1:J){
    a[j] ~ dnorm (a.hat[j], tau.a)
    a.hat[j] <- g.0 + g.1*u[j]
  }
  g.0 ~ dnorm (0, .0001)
  g.1 ~ dnorm (0, .0001)
  tau.a <- pow(sigma.a, -2)
  sigma.a ~ dunif (0, 100)
}

# run the model
radon_0 <- jags (radon.data, radon.inits, radon.parameters, model_0, n.chains=3, n.iter=
1000, DIC=F)
radon_0 %>% plot
radon_0$BUGSoutput
```

Output

```
Inference for Bugs model at "/var/folders/ld/tp92rb3n3gs4zx9_3_xly5fw0000gn/T//Rtmpb2mP8
a/model116f7a158bf2.txt", fit using jags,
 3 chains, each with 1000 iterations (first 500 discarded)
 n.sims = 1500 iterations saved
        mean  sd 2.5%  25%  50%  75% 97.5% Rhat n.eff
a[1]     1.0 0.2  0.6  0.8  1.0  1.1   1.3    1  1500
a[2]     0.9 0.1  0.7  0.8  0.9  0.9   1.0    1  1500
a[3]     1.4 0.2  1.1  1.3  1.4  1.5   1.7    1  1500
a[4]     1.2 0.2  0.9  1.1  1.2  1.3   1.5    1  1500
a[5]     1.4 0.2  1.1  1.3  1.4  1.5   1.7    1   510
a[80]    1.3 0.1  1.2  1.3  1.3  1.4   1.5    1  1500
a[81]    1.8 0.2  1.4  1.6  1.7  1.9   2.1    1   320
a[82]    1.7 0.2  1.3  1.6  1.7  1.8   2.0    1  1300
a[83]    1.7 0.1  1.4  1.6  1.7  1.8   2.0    1   700
a[84]    1.5 0.1  1.3  1.4  1.5  1.6   1.8    1  1500
a[85]    1.7 0.2  1.3  1.6  1.7  1.8   2.0    1  1500
b       -0.7 0.1 -0.8 -0.7 -0.7 -0.6  -0.5    1  1500
g.0      1.5 0.0  1.4  1.4  1.5  1.5   1.5    1   150
g.1      0.7 0.1  0.5  0.7  0.7  0.8   0.9    1   520
sigma.a  0.2 0.0  0.1  0.1  0.2  0.2   0.3    1   100
sigma.y  0.8 0.0  0.7  0.7  0.8  0.8   0.8    1   250

For each parameter, n.eff is a crude measure of effective sample size,
and Rhat is the potential scale reduction factor (at convergence, Rhat=1).
```

Graph



Based on this model, b=-0.7, g.0=1.5, g.1=0.7, sigma.a=0.2, sigma.y=0.8, which means that radon level measured at the first level was 0.7 lower than that measured at the basement.

Part A

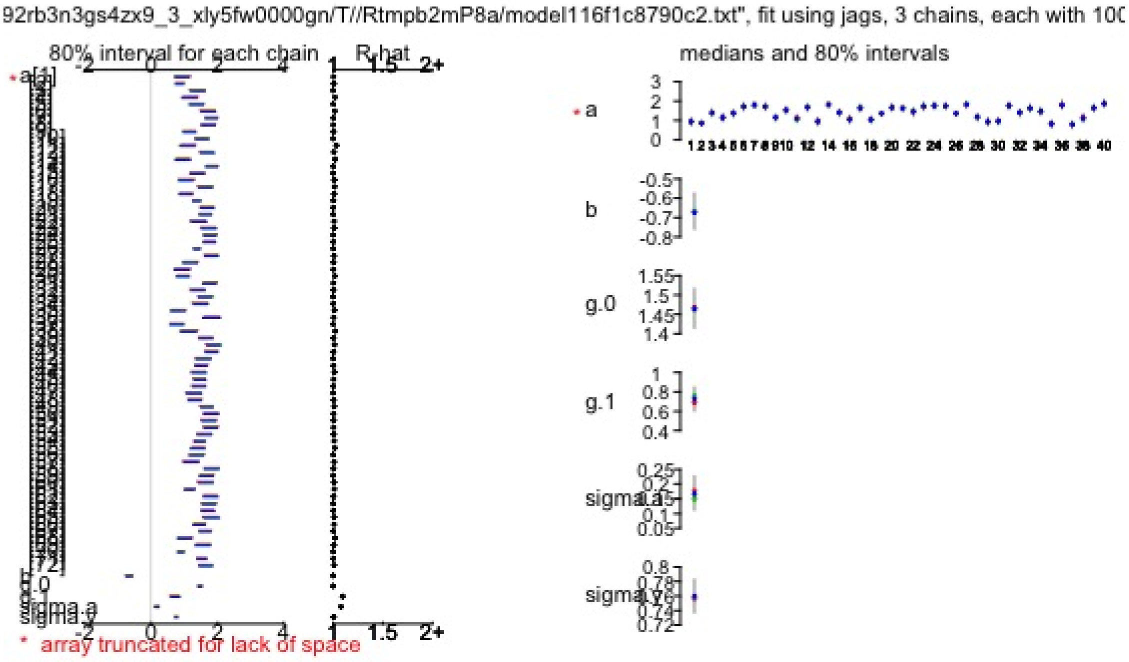
How do the inferences change if you assign normal prior distributions with mean 5 and standard deviation 1000 to the coefficients for floor and uranium.

```
model_a <- function () {
  for (i in 1:n){
    y[i] ~ dnorm (y.hat[i], tau.y)
    y.hat[i] <- a[county[i]] + b*x[i]
  }
  b ~ dnorm (5, 1000^-2)
  tau.y <- pow(sigma.y, -2)
  sigma.y ~ dunif (0, 100)

  for (j in 1:J){
    a[j] ~ dnorm (a.hat[j], tau.a)
    a.hat[j] <- g.0 + g.1*u[j]
  }
  g.0 ~ dnorm (0, .0001)
  g.1 ~ dnorm (5, 1000^-2)
  tau.a <- pow(sigma.a, -2)
  sigma.a ~ dunif (0, 100)
}

# run the model
radon_a <- jags (radon.data, radon.inits, radon.parameters, model_a, n.chains=3, n.iter=
1000, DIC=F)
radon_a %>% plot
radon_a$BUGSoutput
```

Output

```
Inference for Bugs model at "/var/folders/ld/tp92rb3n3gs4zx9_3_xly5fw0000gn/T//Rtmpb2mP8
a/model116f1c8790c2.txt", fit using jags,
 3 chains, each with 1000 iterations (first 500 discarded)
 n.sims = 1500 iterations saved
        mean  sd 2.5%  25%  50%  75% 97.5% Rhat n.eff
a[1]      0.9 0.2  0.6  0.8  0.9  1.0   1.3  1.0   460
a[2]      0.9 0.1  0.7  0.8  0.9  0.9   1.0  1.0   370
a[3]      1.4 0.2  1.1  1.3  1.4  1.5   1.7  1.0  1500
a[4]      1.2 0.2  0.9  1.0  1.1  1.3   1.5  1.0   130
a[80]     1.3 0.1  1.2  1.3  1.3  1.4   1.5  1.0  1500
a[81]     1.7 0.2  1.4  1.6  1.7  1.8   2.1  1.0  1500
a[82]     1.7 0.2  1.3  1.6  1.7  1.8   2.0  1.0  1100
a[83]     1.7 0.1  1.4  1.6  1.7  1.8   2.0  1.0   510
a[84]     1.5 0.1  1.2  1.4  1.5  1.6   1.8  1.0  1300
a[85]     1.7 0.2  1.3  1.6  1.7  1.8   2.0  1.0   470
b        -0.7 0.1 -0.8 -0.7 -0.7 -0.6  -0.5  1.0  1500
g.0       1.5 0.0  1.4  1.4  1.5  1.5   1.5  1.0  1500
g.1       0.7 0.1  0.5  0.7  0.7  0.8   0.9  1.1    29
sigma.a   0.2 0.0  0.1  0.1  0.2  0.2   0.3  1.1    47
sigma.y   0.8 0.0  0.7  0.7  0.8  0.8   0.8  1.0   500

For each parameter, n.eff is a crude measure of effective sample size,
and Rhat is the potential scale reduction factor (at convergence, Rhat=1).
```

Graph



Based on this model, b=-0.7, g.1=0.7, which is the same with model_0. Although we assigned the normal prior distribution with mean 5, the standard deviation of 1000 makes the prior distribution a non-informative one condisering the large value of standard deviation. Thus the results did not show difference as model_0.

Part B

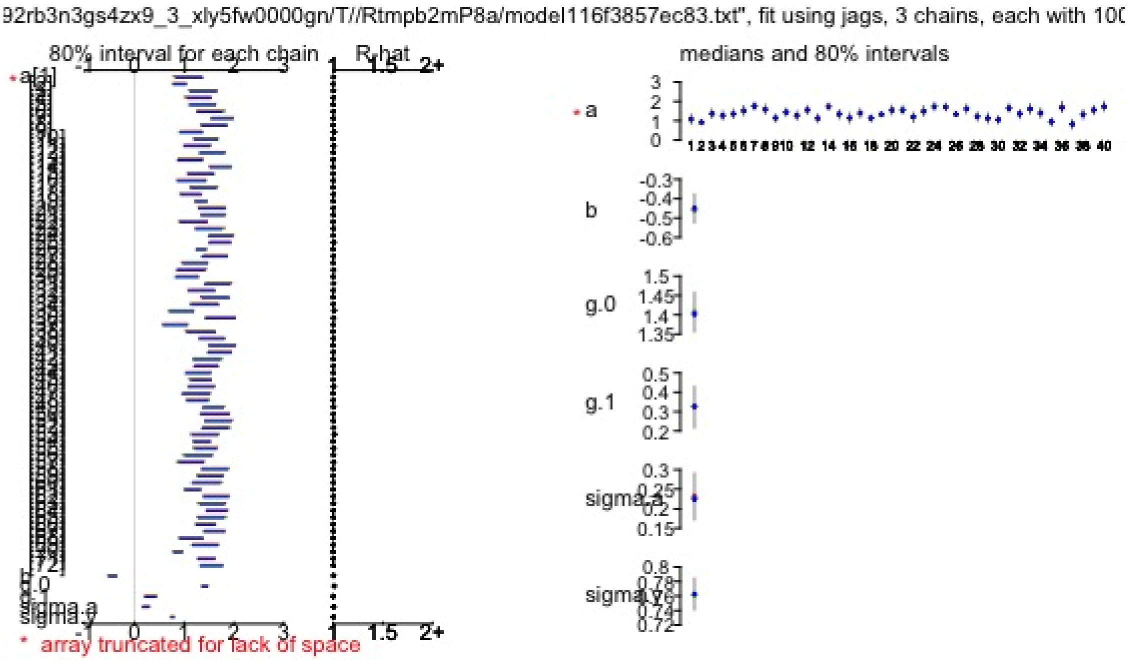
How do the inferences change if you switch to normal prior distributions with mean 0 and standard deviation 0.1?

```
model_b <- function () {
  for (i in 1:n){
    y[i] ~ dnorm (y.hat[i], tau.y)
    y.hat[i] <- a[county[i]] + b*x[i]
  }
  b ~ dnorm (0, 0.1^-2)
  tau.y <- pow(sigma.y, -2)
  sigma.y ~ dunif (0, 100)

  for (j in 1:J){
    a[j] ~ dnorm (a.hat[j], tau.a)
    a.hat[j] <- g.0 + g.1*u[j]
  }
  g.0 ~ dnorm (0, .0001)
  g.1 ~ dnorm (0, 0.1^-2)
  tau.a <- pow(sigma.a, -2)
  sigma.a ~ dunif (0, 100)
}

# run the model
radon_b <- jags (radon.data, radon.inits, radon.parameters, model_b, n.chains=3, n.iter=
1000, DIC=F)
radon_b %>% plot
radon_b$BUGSoutput
```

Output

```
Inference for Bugs model at "/var/folders/ld/tp92rb3n3gs4zx9_3_xly5fw0000gn/T//Rtmpb2mP8
a/model116f3857ec83.txt", fit using jags,
 3 chains, each with 1000 iterations (first 500 discarded)
 n.sims = 1500 iterations saved
        mean  sd 2.5%  25%  50%  75% 97.5% Rhat n.eff
a[1]     1.1 0.2  0.7  0.9  1.1  1.2   1.5    1  1000
a[2]     0.9 0.1  0.7  0.8  0.9  1.0   1.1    1  1500
a[3]     1.4 0.2  1.0  1.2  1.4  1.5   1.8    1   990
a[4]     1.3 0.2  0.9  1.1  1.3  1.4   1.7    1  1100
a[5]     1.4 0.2  1.0  1.2  1.3  1.5   1.7    1  1500
a[80]    1.3 0.1  1.1  1.2  1.3  1.4   1.5    1  1500
a[81]    1.7 0.2  1.3  1.5  1.7  1.8   2.2    1   810
a[82]    1.6 0.2  1.1  1.4  1.6  1.7   2.0    1  1500
a[83]    1.6 0.2  1.2  1.5  1.6  1.7   1.9    1   690
a[84]    1.5 0.2  1.2  1.4  1.5  1.6   1.8    1  1500
a[85]    1.5 0.2  1.0  1.3  1.5  1.6   1.9    1   310
b       -0.5 0.1 -0.6 -0.5 -0.5 -0.4  -0.3    1   420
g.0      1.4 0.0  1.3  1.4  1.4  1.4   1.5    1   530
g.1      0.3 0.1  0.2  0.3  0.3  0.4   0.5    1  1500
sigma.a  0.2 0.0  0.1  0.2  0.2  0.3   0.3    1   430
sigma.y  0.8 0.0  0.7  0.8  0.8  0.8   0.8    1   730

For each parameter, n.eff is a crude measure of effective sample size,
and Rhat is the potential scale reduction factor (at convergence, Rhat=1).
```

Graph



Based on this model, b=-0.5, g.1=0.3, both of them are closer to 0 (the mean of normal prior distributions) compared to model_0 because that model_b has a narrow standard deviation (0.1) of the prior distribution which is a informative prior distribution.

Part C

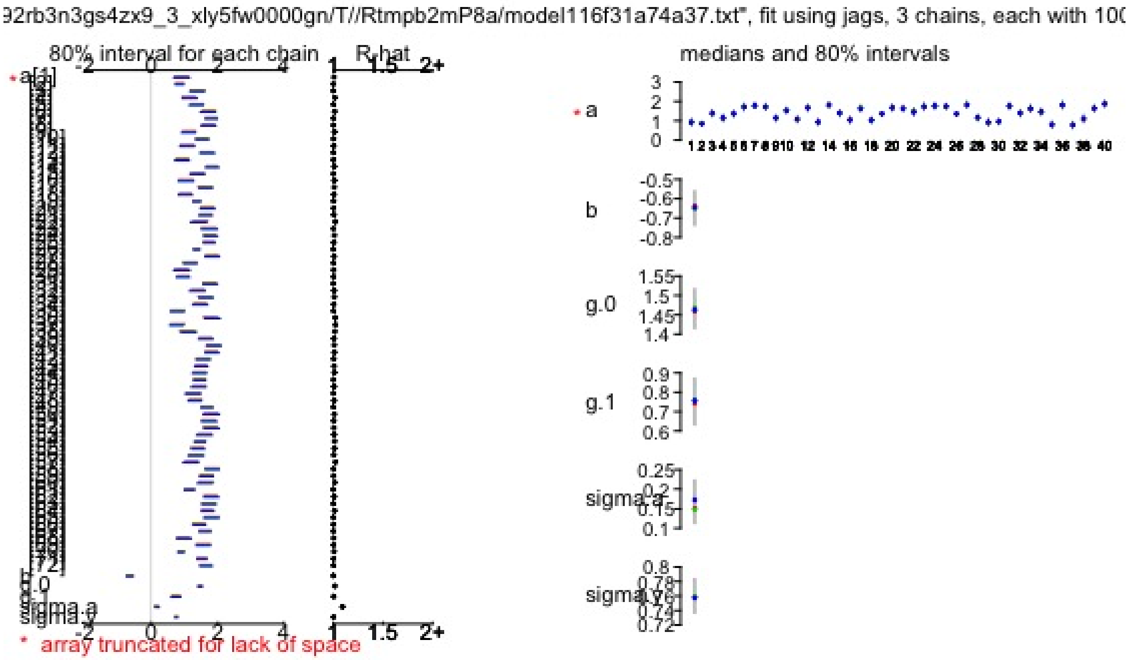
Now try normal prior distributions with mean 5 and standard deviation 1.

```
model_c <- function () {
  for (i in 1:n){
    y[i] ~ dnorm (y.hat[i], tau.y)
    y.hat[i] <- a[county[i]] + b*x[i]
  }
  b ~ dnorm (5, 1^-2)
  tau.y <- pow(sigma.y, -2)
  sigma.y ~ dunif (0, 100)

  for (j in 1:J){
    a[j] ~ dnorm (a.hat[j], tau.a)
    a.hat[j] <- g.0 + g.1*u[j]
  }
  g.0 ~ dnorm (0, .0001)
  g.1 ~ dnorm (5, 1^-2)
  tau.a <- pow(sigma.a, -2)
  sigma.a ~ dunif (0, 100)
}

# run the model
radon_c <- jags (radon.data, radon.inits, radon.parameters, model_c, n.chains=3, n.iter=
1000, DIC=F)
radon_c %>% plot
radon_c$BUGSoutput
```

Output

```
Inference for Bugs model at "/var/folders/ld/tp92rb3n3gs4zx9_3_xly5fw0000gn/T//Rtmpb2mP8
a/model116f31a74a37.txt", fit using jags,
 3 chains, each with 1000 iterations (first 500 discarded)
 n.sims = 1500 iterations saved
        mean  sd 2.5%  25%  50%  75% 97.5% Rhat n.eff
a[1]     0.9 0.2  0.6  0.8  0.9  1.0   1.3  1.0   750
a[2]     0.9 0.1  0.7  0.8  0.9  0.9   1.0  1.0  1500
a[3]     1.4 0.2  1.1  1.3  1.4  1.5   1.7  1.0  1500
a[4]     1.1 0.2  0.9  1.0  1.1  1.2   1.5  1.0   180
a[5]     1.4 0.2  1.1  1.3  1.4  1.5   1.7  1.0  1500
a[80]    1.3 0.1  1.1  1.3  1.3  1.4   1.5  1.0  1500
a[81]    1.7 0.2  1.4  1.6  1.7  1.8   2.1  1.0   360
a[82]    1.7 0.2  1.3  1.6  1.7  1.8   2.0  1.0  1500
a[83]    1.7 0.1  1.5  1.7  1.7  1.8   2.0  1.0   210
a[84]    1.5 0.1  1.2  1.4  1.5  1.6   1.8  1.0  1500
a[85]    1.7 0.2  1.3  1.6  1.7  1.8   2.0  1.0   190
b       -0.6 0.1 -0.8 -0.7 -0.6 -0.6  -0.5  1.0  1000
g.0      1.5 0.0  1.4  1.4  1.5  1.5   1.5  1.0   230
g.1      0.8 0.1  0.6  0.7  0.8  0.8   0.9  1.0   860
sigma.a  0.2 0.0  0.1  0.1  0.2  0.2   0.3  1.1    35
sigma.y  0.8 0.0  0.7  0.7  0.8  0.8   0.8  1.0  1500

For each parameter, n.eff is a crude measure of effective sample size,
and Rhat is the potential scale reduction factor (at convergence, Rhat=1).
```

Graph



Based on this model, b=-0.6, g.1=0.8, both of them are getting closer to 5 compare to model_0 because we assigned a prior distribution with mean of 5, and standard deviation of 1. Such sd provides constrains on the coefficients in some degree.

Part D

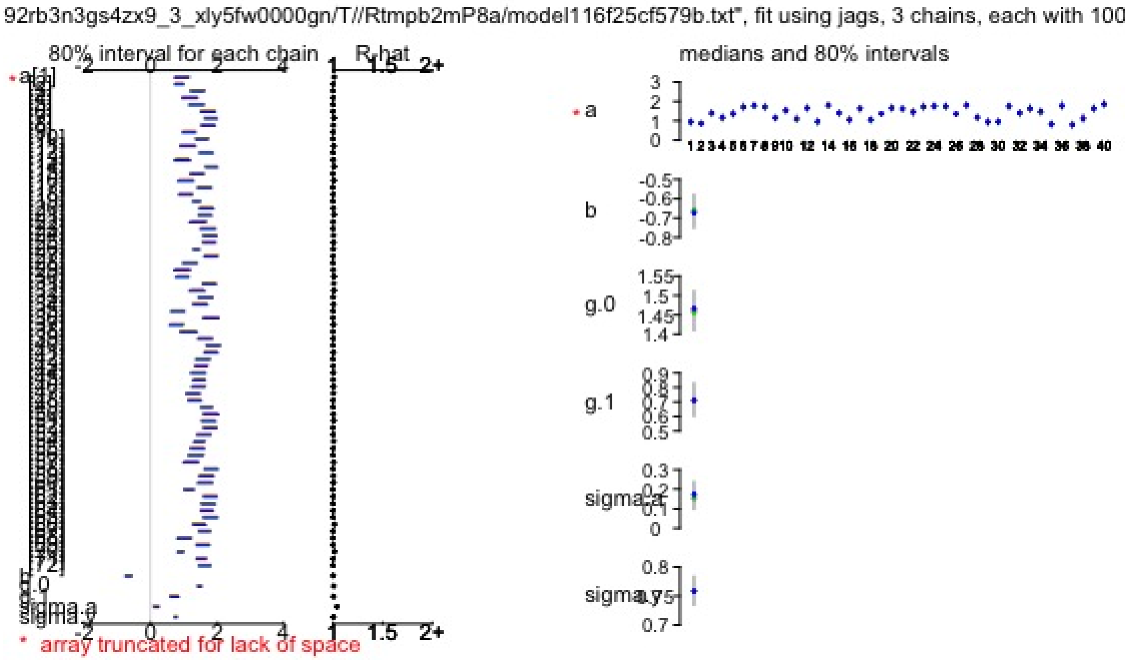
Now try t prior distributions with mean 5, standard deviation 1, and 4 degrees of freedom.

```
model_d <- function () {
  for (i in 1:n){
    y[i] ~ dnorm (y.hat[i], tau.y)
    y.hat[i] <- a[county[i]] + b*x[i]
  }
  b ~ dt (5, 1^-2, 4)
  tau.y <- pow(sigma.y, -2)
  sigma.y ~ dunif (0, 100)

  for (j in 1:J){
    a[j] ~ dnorm (a.hat[j], tau.a)
    a.hat[j] <- g.0 + g.1*u[j]
  }
  g.0 ~ dnorm (0, .0001)
  g.1 ~ dt (5, 1^-2, 4)
  tau.a <- pow(sigma.a, -2)
  sigma.a ~ dunif (0, 100)
}

# run the model
radon_d <- jags (radon.data, radon.inits, radon.parameters, model_d, n.chains=3, n.iter=
1000, DIC=F)
radon_d %>% plot
radon_d$BUGSoutput
```

Output

```
Inference for Bugs model at "/var/folders/ld/tp92rb3n3gs4zx9_3_xly5fw0000gn/T//Rtmpb2mP8
a/model116f25cf579b.txt", fit using jags,
 3 chains, each with 1000 iterations (first 500 discarded)
 n.sims = 1500 iterations saved
        mean  sd 2.5%  25%  50%  75% 97.5% Rhat n.eff
a[1]     0.9 0.2  0.6  0.8  0.9  1.0   1.3    1   600
a[2]     0.9 0.1  0.7  0.8  0.9  0.9   1.0    1   780
a[3]     1.4 0.2  1.1  1.3  1.4  1.5   1.7    1   830
a[4]     1.2 0.2  0.9  1.0  1.2  1.3   1.5    1   790
a[5]     1.4 0.2  1.1  1.3  1.4  1.5   1.7    1  1500
a[80]    1.3 0.1  1.1  1.3  1.3  1.4   1.5    1  1000
a[81]    1.7 0.2  1.4  1.6  1.7  1.8   2.1    1   270
a[82]    1.7 0.2  1.3  1.5  1.7  1.8   2.0    1  1200
a[83]    1.7 0.1  1.5  1.6  1.7  1.8   2.0    1  1500
a[84]    1.5 0.1  1.2  1.4  1.5  1.6   1.8    1   730
a[85]    1.7 0.2  1.3  1.6  1.7  1.8   2.0    1  1500
b       -0.7 0.1 -0.8 -0.7 -0.7 -0.6  -0.5    1   580
g.0      1.5 0.0  1.4  1.4  1.5  1.5   1.5    1   360
g.1      0.7 0.1  0.5  0.7  0.7  0.8   0.9    1   360
sigma.a  0.2 0.1  0.1  0.1  0.2  0.2   0.3    1    71
sigma.y  0.8 0.0  0.7  0.7  0.8  0.8   0.8    1   560

For each parameter, n.eff is a crude measure of effective sample size,
and Rhat is the potential scale reduction factor (at convergence, Rhat=1).
```

Graph



Based on this model, b=-0.7, g.1=0.7, which is the same with model_0. Although we assigned a prior t-distrubution of mean 5 and std 1, the degree of freedom of is relatively small. Thus it makes the prior distribution wide and flat which provides almost no constrains on the coefficients as a noninformative prior distribution.

Part E

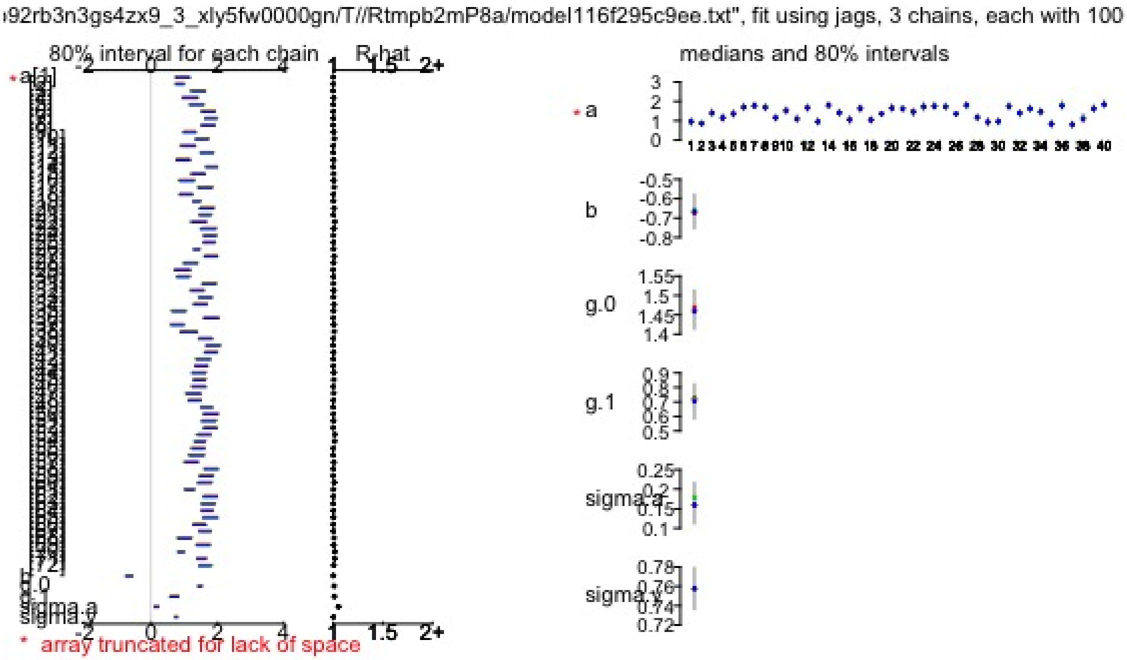
Now try Uniform(−100,100) prior distributions, then Uniform(−1,1) prior distributions.

```
model_e1 <- function () {
  for (i in 1:n){
    y[i] ~ dnorm (y.hat[i], tau.y)
    y.hat[i] <- a[county[i]] + b*x[i]
  }
  b ~ dunif (-100, 100)
  tau.y <- pow(sigma.y, -2)
  sigma.y ~ dunif (0, 100)

  for (j in 1:J){
    a[j] ~ dnorm (a.hat[j], tau.a)
    a.hat[j] <- g.0 + g.1*u[j]
  }
  g.0 ~ dnorm (0, .0001)
  g.1 ~ dunif (-100, 100)
  tau.a <- pow(sigma.a, -2)
  sigma.a ~ dunif (0, 100)
}

# run the model
radon_e1 <- jags (radon.data, radon.inits, radon.parameters, model_e1, n.chains=3, n.ite
r=1000, DIC=F)
radon_e1 %>% plot
radon_e1$BUGSoutput
```

Output

```
Inference for Bugs model at "/var/folders/ld/tp92rb3n3gs4zx9_3_xly5fw0000gn/T//Rtmpb2mP8
a/model116f295c9ee.txt", fit using jags,
 3 chains, each with 1000 iterations (first 500 discarded)
 n.sims = 1500 iterations saved
        mean  sd 2.5%  25%  50%  75% 97.5% Rhat n.eff
a[1]     0.9 0.2  0.6  0.8  0.9  1.1   1.3  1.0  1500
a[2]     0.9 0.1  0.7  0.8  0.9  0.9   1.0  1.0  1500
a[3]     1.4 0.2  1.1  1.3  1.4  1.5   1.7  1.0  1500
a[4]     1.2 0.2  0.9  1.0  1.2  1.3   1.5  1.0   550
a[5]     1.4 0.2  1.1  1.3  1.4  1.5   1.7  1.0  1500
a[80]    1.3 0.1  1.2  1.3  1.3  1.4   1.5  1.0  1500
a[81]    1.7 0.2  1.4  1.6  1.7  1.8   2.1  1.0  1500
a[82]    1.7 0.2  1.3  1.5  1.6  1.8   2.0  1.0   550
a[83]    1.7 0.1  1.5  1.6  1.7  1.8   2.0  1.0   410
a[84]    1.5 0.1  1.2  1.4  1.5  1.6   1.8  1.0  1500
a[85]    1.7 0.2  1.3  1.6  1.7  1.8   2.0  1.0   490
b       -0.7 0.1 -0.8 -0.7 -0.7 -0.6  -0.5  1.0  1200
g.0      1.5 0.0  1.4  1.4  1.5  1.5   1.5  1.0   260
g.1      0.7 0.1  0.5  0.6  0.7  0.8   0.9  1.0   380
sigma.a  0.2 0.0  0.1  0.1  0.2  0.2   0.2  1.1    51
sigma.y  0.8 0.0  0.7  0.7  0.8  0.8   0.8  1.0  1500

For each parameter, n.eff is a crude measure of effective sample size,
and Rhat is the potential scale reduction factor (at convergence, Rhat=1).
```

Graph

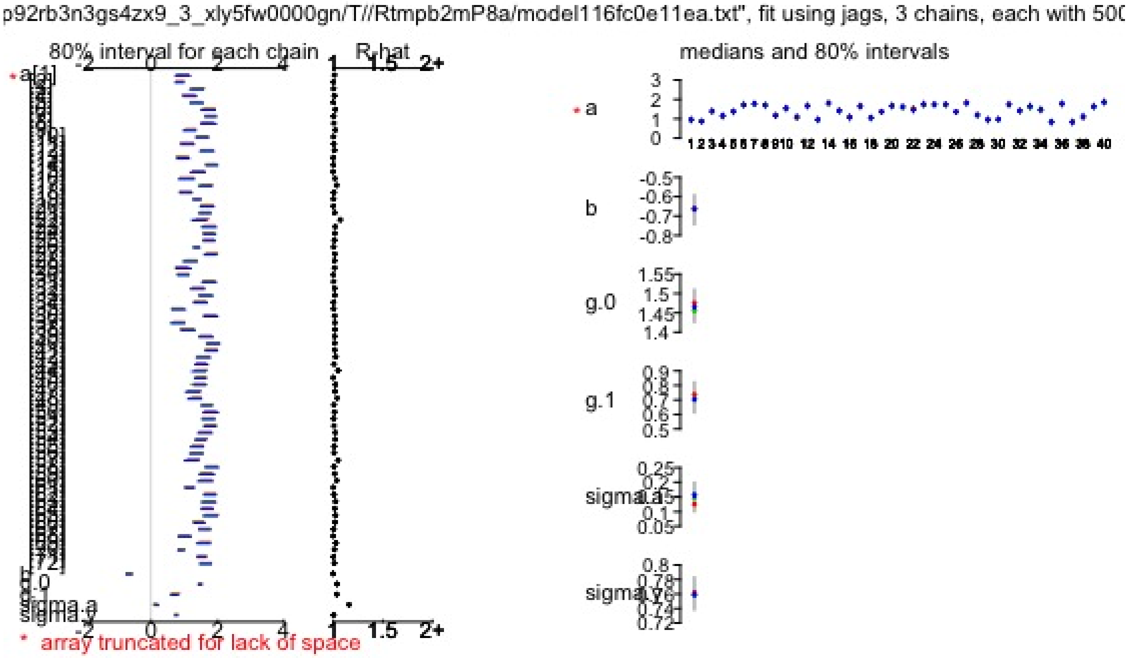


```
radon_inits_e2 <- function (){
  list (a=rnorm(J),
        b=runif(1, -1, 1),
        g.0=rnorm(1),
        g.1=runif(1, -1, 1),
        sigma.y=runif(1),
        sigma.a=runif(1))
}

model_e2 <- function () {
  for (i in 1:n){
    y[i] ~ dnorm (y.hat[i], tau.y)
    y.hat[i] <- a[county[i]] + b*x[i]
  }
  b ~ dunif (-1, 1)
  tau.y <- pow(sigma.y, -2)
  sigma.y ~ dunif (0, 100)

  for (j in 1:J){
    a[j] ~ dnorm (a.hat[j], tau.a)
    a.hat[j] <- g.0 + g.1*u[j]
  }
  g.0 ~ dnorm (0, .0001)
  g.1 ~ dunif (-1, 1)
  tau.a <- pow(sigma.a, -2)
  sigma.a ~ dunif (0, 100)
}

# run the model
radon_e2 <- jags (radon.data, radon_inits_e2, radon.parameters, model_e2, n.chains=3, n.
iter=500, DIC=F)
radon_e2 %>% plot
radon_e2$BUGSoutput
```

Output

```
Inference for Bugs model at "/var/folders/ld/tp92rb3n3gs4zx9_3_xly5fw0000gn/T//Rtmpb2mP8
a/model116fc0e11ea.txt", fit using jags,
 3 chains, each with 500 iterations (first 250 discarded)
 n.sims = 750 iterations saved
        mean  sd 2.5%  25%  50%  75% 97.5% Rhat n.eff
a[1]     1.0 0.1  0.6  0.9  1.0  1.1   1.2  1.0   750
a[2]     0.9 0.1  0.7  0.8  0.9  0.9   1.0  1.0   600
a[3]     1.4 0.1  1.1  1.3  1.4  1.5   1.7  1.0   420
a[4]     1.1 0.1  0.9  1.0  1.1  1.2   1.4  1.0   180
a[5]     1.4 0.1  1.1  1.3  1.4  1.5   1.7  1.0   750
a[80]    1.3 0.1  1.2  1.3  1.3  1.4   1.5  1.0   750
a[81]    1.7 0.2  1.4  1.6  1.7  1.8   2.0  1.0   750
a[82]    1.7 0.2  1.4  1.6  1.7  1.8   2.0  1.0   140
a[83]    1.7 0.1  1.5  1.6  1.7  1.8   2.0  1.0   120
a[84]    1.5 0.1  1.2  1.4  1.5  1.6   1.7  1.0   750
a[85]    1.7 0.2  1.3  1.6  1.7  1.8   2.0  1.0   130
b       -0.7 0.1 -0.8 -0.7 -0.7 -0.6  -0.6  1.0   750
g.0      1.5 0.0  1.4  1.4  1.5  1.5   1.5  1.0    56
g.1      0.7 0.1  0.5  0.7  0.7  0.8   0.9  1.0    70
sigma.a  0.1 0.0  0.1  0.1  0.1  0.2   0.2  1.2    23
sigma.y  0.8 0.0  0.7  0.7  0.8  0.8   0.8  1.0   450

For each parameter, n.eff is a crude measure of effective sample size,
and Rhat is the potential scale reduction factor (at convergence, Rhat=1).
```

Graph



Based on this two models, b=-0.7, g.1=0.7, which is the same with model_0. That's because both of these two are within (–1,1) and (–100,100). If b and g.1 are beyond these two range in the original model_0, the results here will be much different from model_0.

Ch16.3 with varying slopes

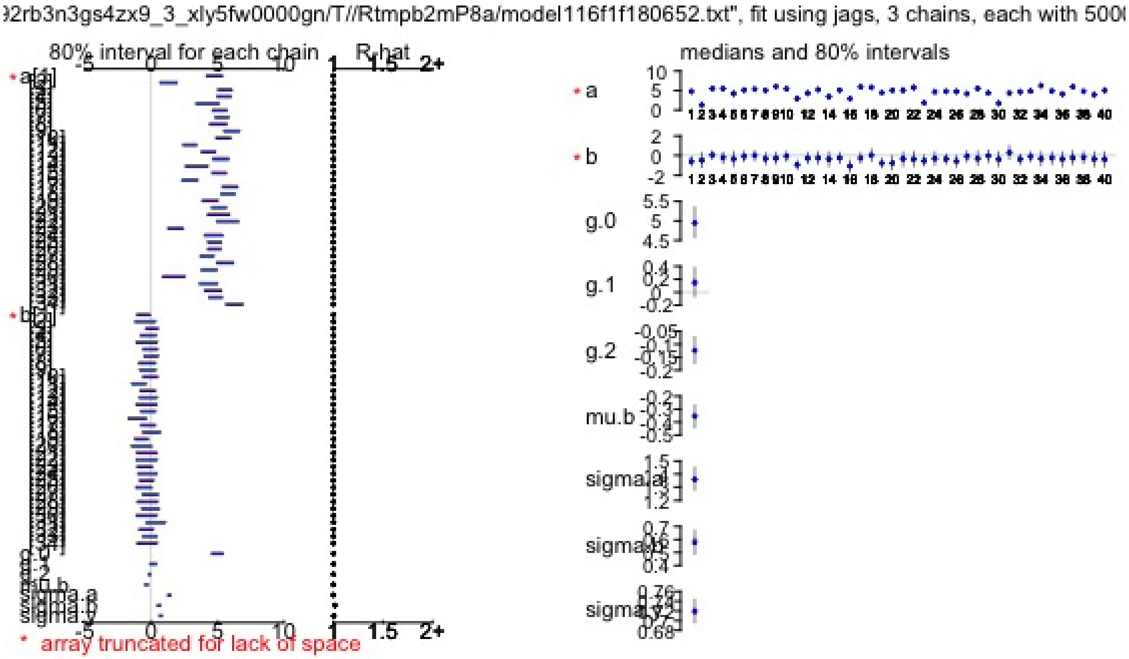
```
# using JAGS
CD4PCT <- hiv_data_2$CD4PCT
n <- length(CD4PCT)
y <- sqrt(CD4PCT)
x <- hiv_data_2$time
P <- hiv_data_2[, c('newpid', 'treatmnt', 'baseage')] %>% unique %>% select(., -(newpi
d))
id.name <- as.vector(hiv_data_2$newpid)
uniq <- unique(id.name)
J <- length(uniq)
id <- rep (NA, J)
for (i in 1:J){
  id[id.name==uniq[i]] <- i
}

# model
model_3 <- function(){
  for (i in 1:n){
    y[i] ~ dnorm (y.hat[i], tau.y)
    y.hat[i] <- a[id[i]] + b[id[i]]*x[i]
  }
  tau.y <- pow(sigma.y, -2)
  sigma.y ~ dunif(0, 100)
  for (j in 1:J){
    a[j] ~ dnorm (a.hat[j], tau.a)
    b[j] ~ dnorm (mu.b, tau.b)
    a.hat[j] <- g.0 + g.1*P[j, 1] + g.2*P[j, 2]
  }
  g.0 ~ dnorm (0, .0001)
  g.1 ~ dnorm (0, .0001)
  g.2 ~ dnorm (0, .0001)
  mu.b ~ dnorm (0, .0001)
  tau.a <- pow(sigma.a, -2)
  tau.b <- pow(sigma.b, -2)
  sigma.a ~ dunif (0, 100)
  sigma.b ~ dunif (0, 100)
}

# hiv.inits and hiv.parameters
hiv_data_list_3 <- list ("n", "J", "y", "id", "x", "P")
hiv_inits_3 <- function () {
  list (a=rnorm(J),
        b=rnorm(J),
        g.0=rnorm(1),
        g.1=rnorm(1),
        g.2=rnorm(1),
        mu.b = rnorm(1),
        sigma.y=runif(1),
        sigma.a=runif(1),
        sigma.b=runif(1))
}
hiv_parameters_3 <- c ("a", "b", "g.0", "g.1", "g.2", "mu.b", "sigma.y", "sigma.a", "sig
ma.b")
```

```
# run the model
hiv_bugs_3 <- jags(hiv_data_list_3, hiv_inits_3, hiv_parameters_3, model_3, n.chains=3,
 n.iter=5000, DIC=F)
hiv_bugs_3 %>% plot
hiv_bugs_3$BUGSoutput
```

Output

```
Inference for Bugs model at "/var/folders/ld/tp92rb3n3gs4zx9_3_xly5fw0000gn/T//Rtmpb2mP8
a/model116f1f180652.txt", fit using jags,
 3 chains, each with 5000 iterations (first 2500 discarded), n.thin = 2
 n.sims = 3750 iterations saved
        mean  sd 2.5%  25%  50%  75% 97.5% Rhat n.eff
a[1]     4.7 0.4  3.9  4.4  4.8  5.1   5.6    1  3800
a[2]     1.3 0.5  0.4  1.0  1.3  1.6   2.3    1  3300
a[3]     5.5 0.4  4.8  5.3  5.5  5.8   6.3    1  2500
a[4]     5.5 0.4  4.7  5.2  5.5  5.7   6.2    1  3800
a[5]     4.2 0.6  3.0  3.8  4.2  4.7   5.5    1  3800
a[245]   5.5 0.6  4.2  5.0  5.5  5.9   6.7    1  3800
a[246]   5.7 0.5  4.7  5.4  5.7  6.0   6.6    1  3800
a[247]   4.0 0.6  2.8  3.6  4.0  4.4   5.3    1  3800
a[248]   4.6 0.5  3.7  4.3  4.6  4.9   5.5    1  3800
a[249]   3.9 0.6  2.7  3.5  3.9  4.3   5.1    1  3800
a[250]   3.9 0.5  2.9  3.5  3.9  4.2   4.9    1  3400
b[1]    -0.6 0.4 -1.3 -0.8 -0.6 -0.3   0.1    1  1800
b[2]    -0.5 0.6 -1.6 -0.8 -0.4 -0.1   0.7    1  1600
b[3]     0.1 0.3 -0.6 -0.2  0.1  0.3   0.7    1  3800
b[4]    -0.2 0.4 -1.1 -0.5 -0.2  0.1   0.7    1  3800
b[5]    -0.3 0.6 -1.5 -0.7 -0.3  0.0   0.8    1  3800
b[245]  -0.4 0.6 -1.5 -0.7 -0.4  0.0   0.8    1  3800
b[246]  -0.3 0.6 -1.5 -0.7 -0.3  0.1   0.8    1  3800
b[247]  -0.4 0.6 -1.5 -0.7 -0.4  0.0   0.8    1  3400
b[248]  -0.4 0.6 -1.6 -0.8 -0.4 -0.1   0.6    1  3800
b[249]  -0.3 0.6 -1.5 -0.7 -0.3  0.0   0.8    1  1300
b[250]  -0.2 0.5 -1.2 -0.6 -0.2  0.2   0.9    1  3800
g.0      5.0 0.3  4.4  4.7  4.9  5.2   5.6    1  3800
g.1      0.2 0.2 -0.2  0.0  0.2  0.3   0.5    1  2100
g.2     -0.1 0.0 -0.2 -0.2 -0.1 -0.1   0.0    1  3800
mu.b    -0.4 0.1 -0.5 -0.4 -0.4 -0.3  -0.2    1  2800
sigma.a  1.4 0.1  1.2  1.3  1.4  1.4   1.5    1  2900
sigma.b  0.6 0.1  0.4  0.5  0.6  0.6   0.7    1   190
sigma.y  0.7 0.0  0.7  0.7  0.7  0.7   0.8    1   540

For each parameter, n.eff is a crude measure of effective sample size,
and Rhat is the potential scale reduction factor (at convergence, Rhat=1).
```

Graph



Based on this model, we can see that it is a varying-intercept, varying-slope model. The result of this model does not change much as compared with the varying-intercept-only model.

Ch17.5

Models with unequal variances: use Bugs to fit the model you set up for the age-guessing data in Exercise 13.4.

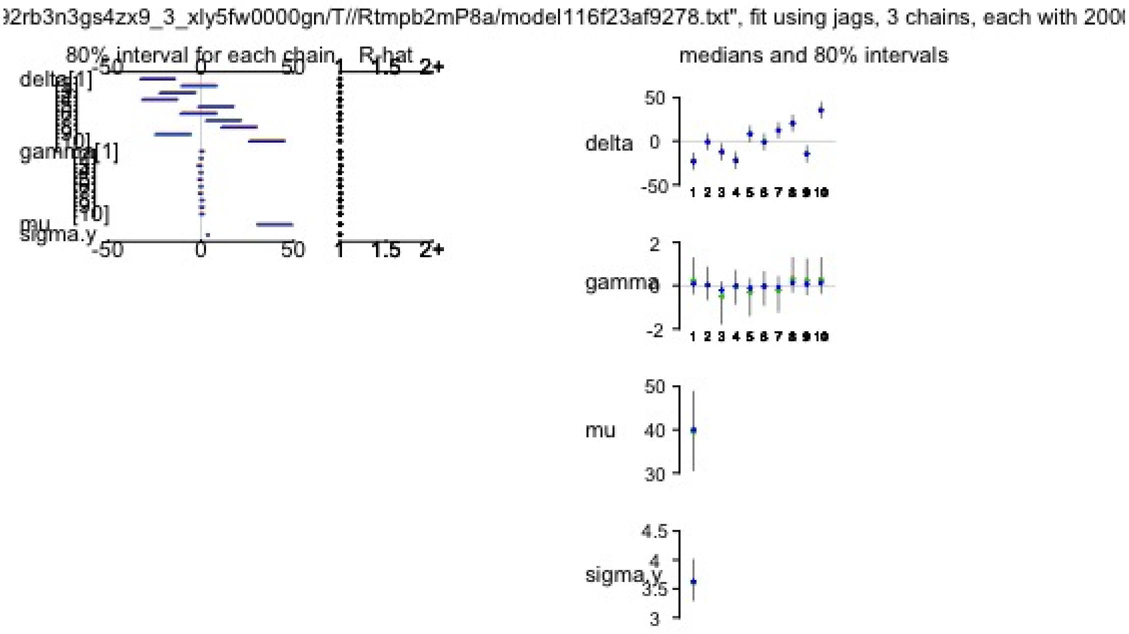
```
# import dataset
age_data_raw <- read.csv('age_guessing.csv', header = T, row.names = 'Group', sep = ';')

# convert dataset
age_data <- as.data.frame(t(age_data_raw[,c(1:10)]))
age_data$photoid<-seq.int(nrow(age_data))
Trueage_data <- age_data[,c(11,12)]
age_data <- age_data[,-11]
age_reshape <-reshape(age_data, direction="long", varying=list(names(age_data)[1:10]),
 v.names="error", idvar=c("photoid"), timevar="groupid")
age_reshape <- cbind(age_reshape,Trueage_data)
age_reshape <- age_reshape[,-5]
colnames(age_reshape) <- c('photoid', 'groupid','error', 'trueage')
age_reshape$guessedage <-age_reshape$error + age_reshape$trueage

# using JAGS
age <- age_reshape$guessedage
n <- length(age)
y <- age
groupid <- as.vector(age_reshape$groupid)
photoid <- as.vector(age_reshape$photoid)
n.groupid <- max(groupid)
n.photoid <- max(photoid)

model_age <- function(){
  for (i in 1:n){
    y[i] ~ dnorm (y.hat[i], tau.y)
    y.hat[i] <- mu + gamma[groupid[i]] + delta[photoid[i]]
  }
  mu ~ dnorm (0, .0001)
  tau.y <- pow(sigma.y, -2)
  sigma.y ~ dunif (0, 100)
  for (j in 1:n.groupid){
    gamma[j] ~ dnorm (0, tau.gamma)
  }
  tau.gamma <- pow(sigma.gamma, -2)
  sigma.gamma ~ dunif (0, 100)
  for (k in 1:n.photoid){
    delta[k] ~ dnorm (0, tau.delta)
  }
  tau.delta <- pow(sigma.delta, -2)
  sigma.delta ~ dunif (0, 100)
}

# hiv.inits and hiv.parameters
age_data_list <- list ("n", "y", "groupid", "n.groupid", "photoid", "n.photoid")
age.inits <- function (){
  list (mu=rnorm(1),
        sigma.delta=runif(1),
        sigma.gamma=runif(1),
        sigma.y=runif(1))
}
age.parameters <- c("mu", "gamma", "delta","sigma.y")
```

```
# run the model
age_bugs <- jags(age_data_list, age.inits, age.parameters, model_age, n.chains=3, n.iter
=2000, DIC=F)
age_bugs %>% plot
age_bugs$BUGSoutput
```

output

```
Inference for Bugs model at "/var/folders/ld/tp92rb3n3gs4zx9_3_xly5fw0000gn/T//Rtmpb2mP8
a/model116f23af9278.txt", fit using jags,
 3 chains, each with 2000 iterations (first 1000 discarded)
 n.sims = 3000 iterations saved
           mean  sd  2.5%   25%   50%   75% 97.5% Rhat n.eff
delta[1]  -23.2 7.5 -38.2 -27.8 -23.0 -18.6  -8.2    1  2700
delta[2]   -0.8 7.5 -15.9  -5.5  -0.9   3.8  13.9    1  2000
delta[3]  -12.5 7.5 -27.6 -17.2 -12.4  -8.0   2.4    1  1900
delta[4]  -22.0 7.5 -37.0 -26.7 -22.0 -17.4  -7.1    1  2600
delta[5]    8.2 7.5  -6.9   3.5   8.2  12.7  23.0    1  2400
delta[6]   -1.1 7.5 -16.3  -5.8  -1.0   3.5  13.7    1  2400
delta[7]   12.3 7.5  -2.8   7.7  12.2  16.9  26.7    1  3000
delta[8]   20.7 7.5   5.9  16.1  20.7  25.3  35.2    1  2400
delta[9]  -14.9 7.5 -29.7 -19.6 -14.7 -10.2  -0.1    1  2500
delta[10]  35.6 7.5  20.4  30.9  35.7  40.3  50.4    1  2600
gamma[1]    0.3 0.7  -0.9  -0.1   0.2   0.7   2.0    1  1500
gamma[2]    0.1 0.6  -1.3  -0.3   0.0   0.4   1.5    1  3000
gamma[3]   -0.6 0.8  -2.5  -1.0  -0.4   0.0   0.6    1   330
gamma[4]   -0.1 0.7  -1.5  -0.4   0.0   0.3   1.3    1  3000
gamma[5]   -0.4 0.7  -2.1  -0.8  -0.2   0.0   0.8    1  1000
gamma[6]   -0.1 0.7  -1.5  -0.4   0.0   0.2   1.2    1  3000
gamma[7]   -0.3 0.7  -2.0  -0.6  -0.2   0.1   0.9    1  1100
gamma[8]    0.4 0.7  -0.7   0.0   0.2   0.7   2.1    1   590
gamma[9]    0.3 0.7  -1.0  -0.1   0.2   0.6   2.0    1   580
gamma[10]   0.4 0.7  -0.8  -0.1   0.2   0.7   2.0    1   680
mu         39.7 7.4  25.1  35.2  39.6  44.3  54.6    1  3000
sigma.y     3.6 0.3   3.1   3.4   3.6   3.8   4.2    1  1200

For each parameter, n.eff is a crude measure of effective sample size,
and Rhat is the potential scale reduction factor (at convergence, Rhat=1).
```

Graph



This is a non-nested model. From the graph we can find that this non-nested model is unstable. This may be due to the small sample size of the present dataset.